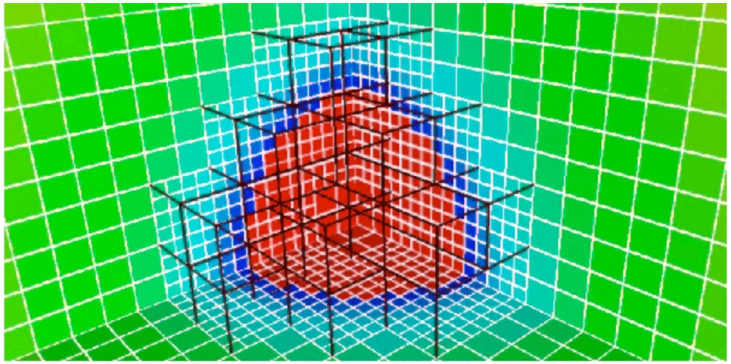
Structured Adaptive Mesh Refinement Applications

VACET Kickoff Meeting
1/31/2007

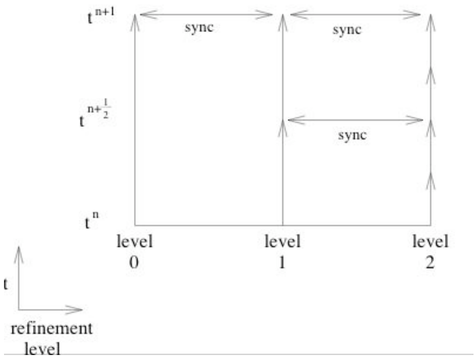
Brian Van Straalen

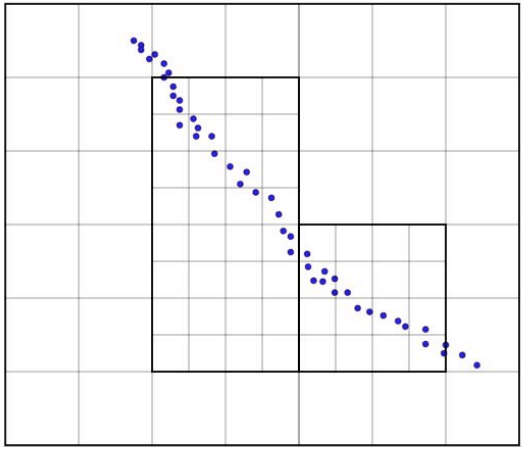
Block-Structured Local Refinement

• Refined regions are organized into rectangular patches.



• Refinement in time as well as in space for time-dependent problems.
• Local refinement can be applied to any structured-grid data, such as bin-sorted particles.





Stakeholders

- **APDEC (SciDAC2 CET)**
  - Combustion (J. Bell, M. Day LBNL)
  - MHD for tokomaks (R. Samtaney PPPL)
  - Wakefield accelerators (W. Mori, E. Esarey LBNL)
  - AMR Elliptic Benchmarks (PERI, TOPS)
  - Astrophysics, Supernovae (J. Bell, S. Woosley/M.Zingale)
- **ANAG (P. Colella, LBNL)**
  - Edge Simulation Laboratory: Edge plasma project (LLNL, LBNL)
    - Tempest (now built on Chombo). New ESL code (5D + 4D)
  - Cosmology: AMR fluid model + AMR PIC code (F. Miniati, ETH)
  - General numerical algorithms research (P. Colella et al, LBNL)
    - MLC, Level Set Methods, Higher-order AMR
  - Microfluidics (D. Trebotich). Electric breakdown/arcing (G. Miller)
  - Ocean internal waves, Estuary modeling (M. Barad Stanford)
- **Larger structured grid AMR Community:**
  - SAMRAI (LLNL)
  - Flash (Chicago)
  - ENZO (Norman UCSD, Abel SLAC)

Current Analysis Tools

- AMRVis
- ChomboVis
- BoxTools stand-alone one-off programs
  - Includes inserting extra post-processing code into CCSE-derived codes and putting computed fields into plotfiles.
- Tecplot and AVS
  - Capable of sophisticated visualization
  - Requires flattening data into fully unstructured, or uniform grid, format

Type 1A Supernovae Theory

- Thermonuclear explosion of a carbon oxygen white dwarf.
- In a white dwarf, electrons provide the needed pressure.
- If the white dwarf is in a binary system and accreting material from it’s partner, then:
  - Mass goes up-> Pressure goes up-> temperature goes up
  - Nuclear burning begins
  - Heating -> convection (about 100 years of convection period)
  - Eventually, cooling through convection can no longer balance the heat generation through reaction. Simmering becomes ignition
  - Initially, flame front is a deflagration (subsonic), allowing star to expand. The flame front accelerates, might become detonation.
  - Finally, the star explodes (roughly 1 second from ignition to explosion)
- The fact that all 1A Supernovae begin with approximately the same mass explains their similarities.

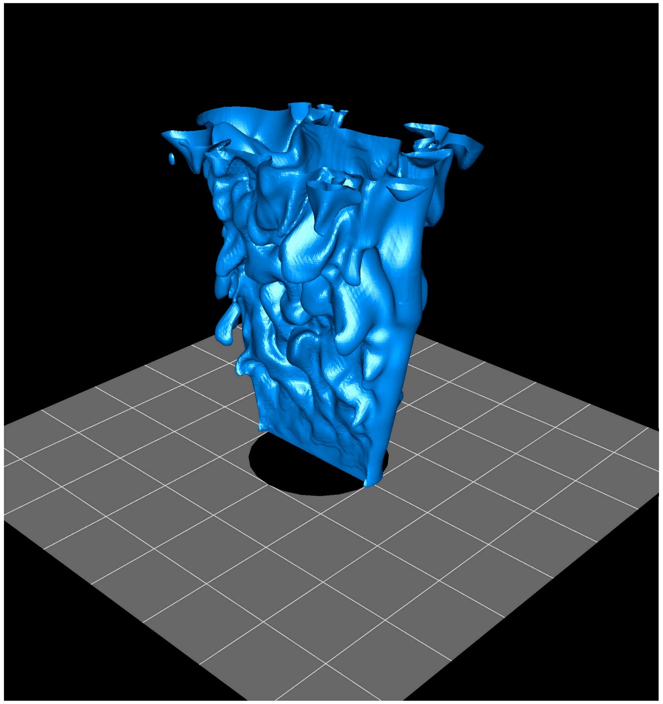
Supernovae Science Goals

- Study conditions leading up to SN 1a ignition
- Model deflagration stage of SN 1a
  - Taking delayed detonation theory beyond curve fitting.
  - Think characterizing many many different RT-instabilities
- X-ray bursts
- Improved methodology for computation of spectra to compare models with observation
- Current Tools used:
  - AMRVis (mostly serial workstation vis)
    - Slices with contours, graphs, box viewing. Mostly debugging.
  - BoxTools postprocessing -> Tecplot
  - Reduction of extremely complex fluid dynamics to simple XY plots.

Combustion

Algorithm Components for low-Mach number combustion algorithm

- Data: Cell-centered AMR grids (3 to 4 levels of refinement)
- Advection: Explicit Godunov with projected velocities
- Diffusion: Crank-Nicolson
- Projection: Density-weighted projection for elliptic constraint
- Chemistry: Stiff ODE integrator (VODE)

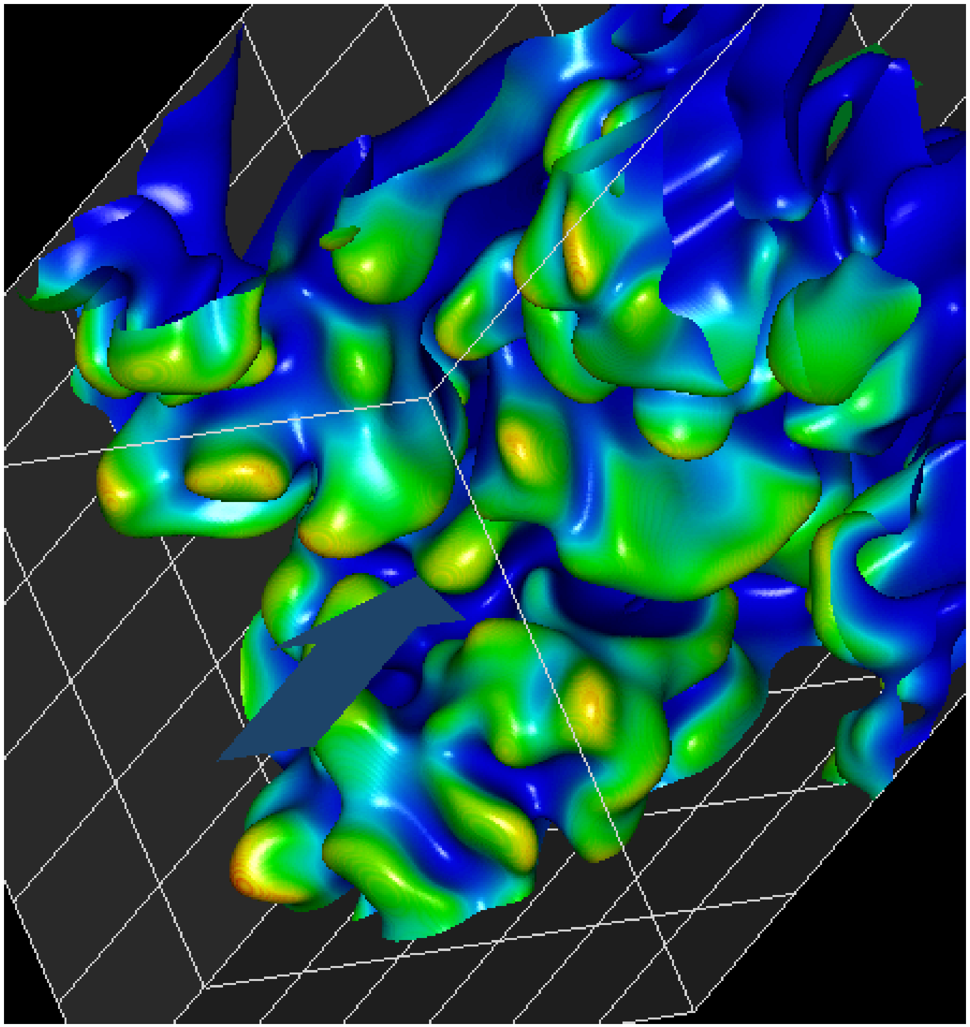


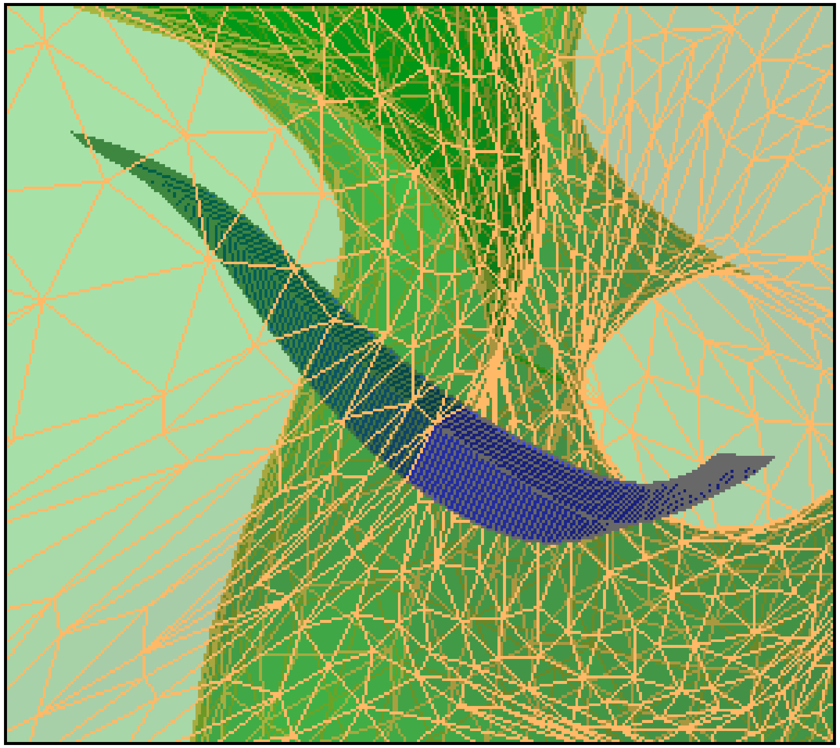
- This will be the main **Embedded Boundary** application in **APDEC**.

Analysis:

- Local "flame surface" based statistics
- differential geometry of implicit surfaces
- volume- and surface-integrals of spatial-temporal regions defined near/on isopleths of field quantities.
- Arbitrary probability distributions (weighted, conditional), based on quantities derived using extensive run-specific physics data.
- Lagrangian analysis along time-dependent vector fields.

•Triangulation extraction based on derived scalar level set (in this case, rate of H2 reduction)





•Integration of volumes normal to active surface to determine flame regions.
•Requires efficient query between unstructured extraction and AMR field data.

Future Directions:

- Detailed transport/chemistry -> Larger runs, harder analysis
  - Pollutant generation in premixed flame systems
  - Higher-order transfer (binary diffusion, Dufour/Soret transport)
  - Complete configurations -> Validation, analysis, geometry
- Realistic inflowing turbulence
- High pressure, temperature
  - Closed combustor
- Geometry -> Visualization/analysis, algorithms
  - Multiple burners
  - Vessel geometry
  - Acoustics

- Data Management:
  - Current research generates O(5 TB) raw data per study
  - Requires frequent repeated access of subsets.
  - This will increase significantly in the next couple of years.
  - Current processing stream is demand-driven, but entire dataset must be resident on disk.

- Performance:

  I/O, parallelism, node performance and debugging are extremely difficult for O(100 - 1000) processors.

  We can only imagine how much harder this will be for O(10000 - 100000) processors!

- Remote :

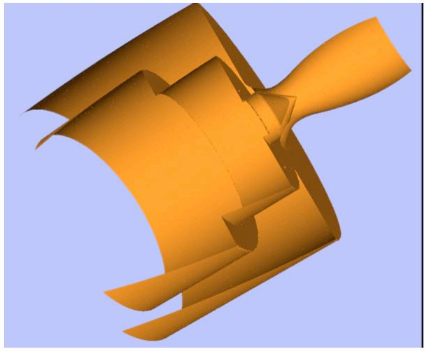
  We have found very few useful remote data subsetting approaches based on high-performance visualization.

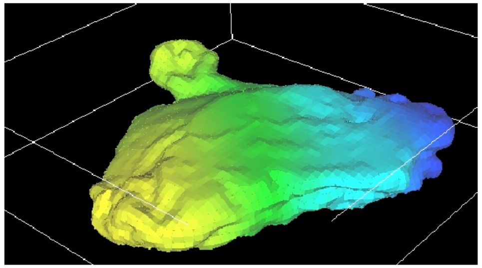
  We must compute on subsets, not just display them.

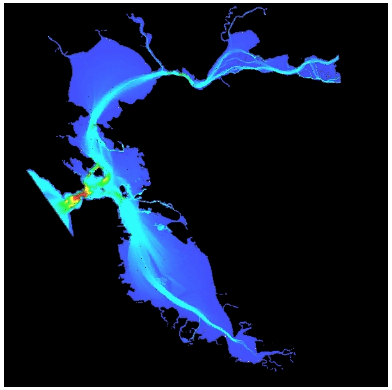
  To date we write our own extractions and view them locally with Tecplot/IDL/etc.

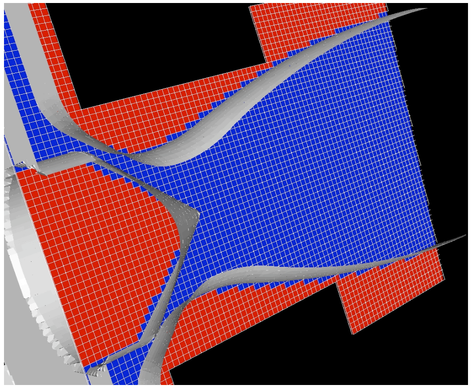
Embedded Boundaries

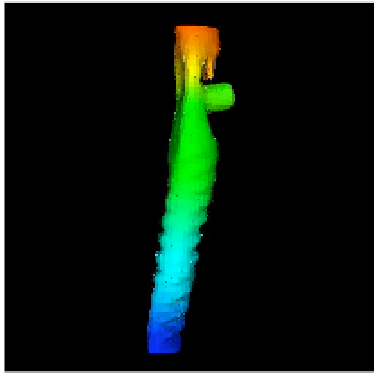
- Development of a variety of grid generation tools.
  - Interface to Cart3D's grid generation package provides an interface to CAD representations of geometry.
  - Implicit function (level set, signed distance function) grid generator

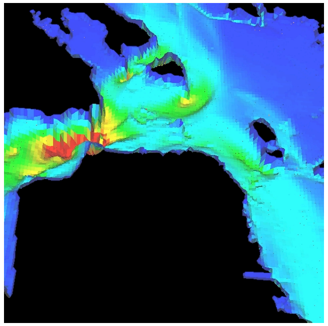






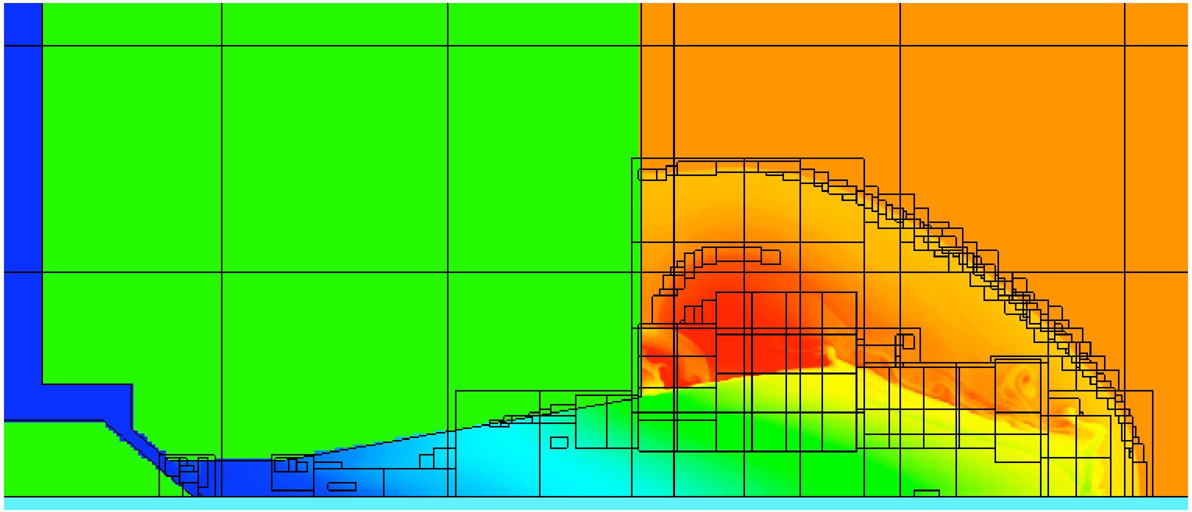


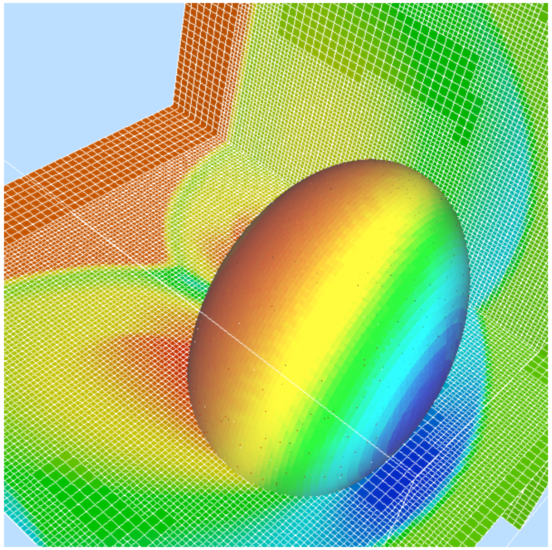




Embedded Boundary Milestones

- Extension of layered software framework to embedded boundary methods.
  - Data holders, interlevel libraries.
  - Robust and accurate hyperbolic, elliptic solvers.
  - Beginning to develop complete applications.

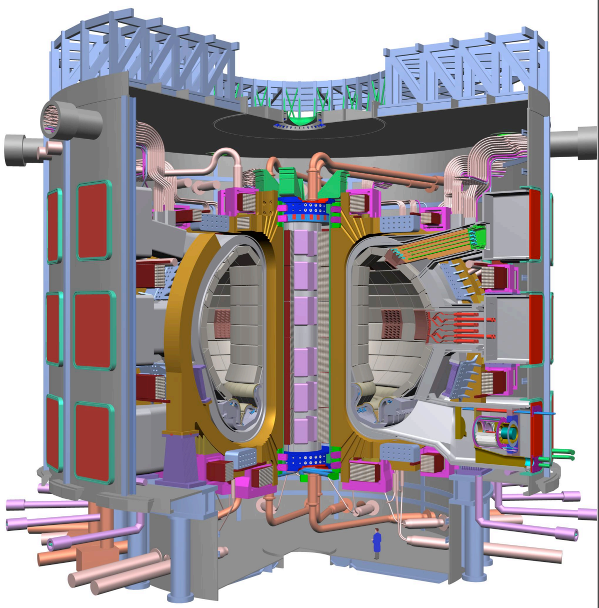




Embedded Boundary Visualization

- Currently visualization is based on a Volume-of-Fluid representation, while simulation is using a cut-cell.
- ChomboVis
  - Flooded contours of slices
  - Debugging: spreadsheets
  - Does not handle multi-volume cut cells
- Auxiliary variables computed inline with simulation code and added to plotfile.
- Future needs
  - Verification and Validation. Matching simulation with experimental rigs (LBNL)
  - Smarter spreadsheets that are EB-aware.
  - Projections of field data onto EB surface with some accuracy.

Scientific Objectives for Fusion Under APDEC



- Two fusion applications have been identified under APDEC
  - <u>Pellet injection</u>
    - Description: In this process a frozen Deuterium pellet is Injected into a tokamak. Experimentally it is known that Injection of frozen pellets is a viable method of fueling a tokamak. ITER is designed to have nine pellet injectors to refuel the burning plasma experiment
    - Identifying the physical processes in pellet injection
    - Understanding the MHD mechanisms for mass redistrbution in tokamaks
    - Causes of cloud striations in pellet injection experiments
  - <u>Edge Localized Modes</u> (ELMs)
    - Description: ELMs are short periodic perturbations at the plasma edge which lead to particle and energy loss from the main plasma during H-mode operation
    - Understanding the underlying MHD mechanisms and precursors leading to ELMs
    - Quantifying the heat loads and frequency of ELMs
  - Both phenomena are related - pellet injection can cause ELMs

- <u>Overarching objective</u>: Develop a comprehensive simulation capability for pellet injection and ELMs in tokamaks (esp. ITER) with adaptive mesh refinement for spatial resolution and implicit treatment of the stiff MHD modes

Pellet Injection - HFS vs. LFS

B_T = 0.375T

n_0=1.5£ $10^{19}/m^3$

T_{e1}=1.3Kev

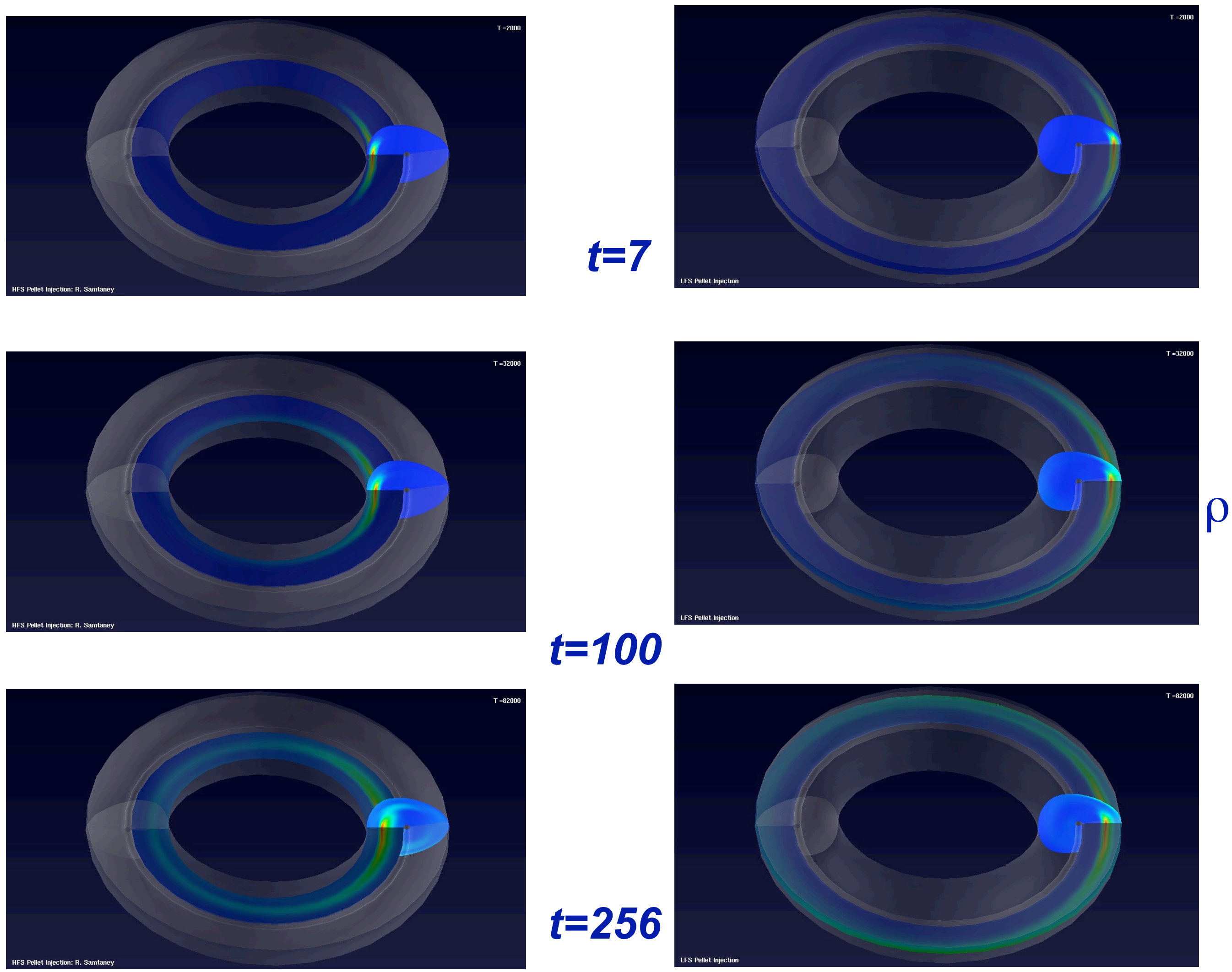
β=0.05

R_0=1m, a=0.3 m

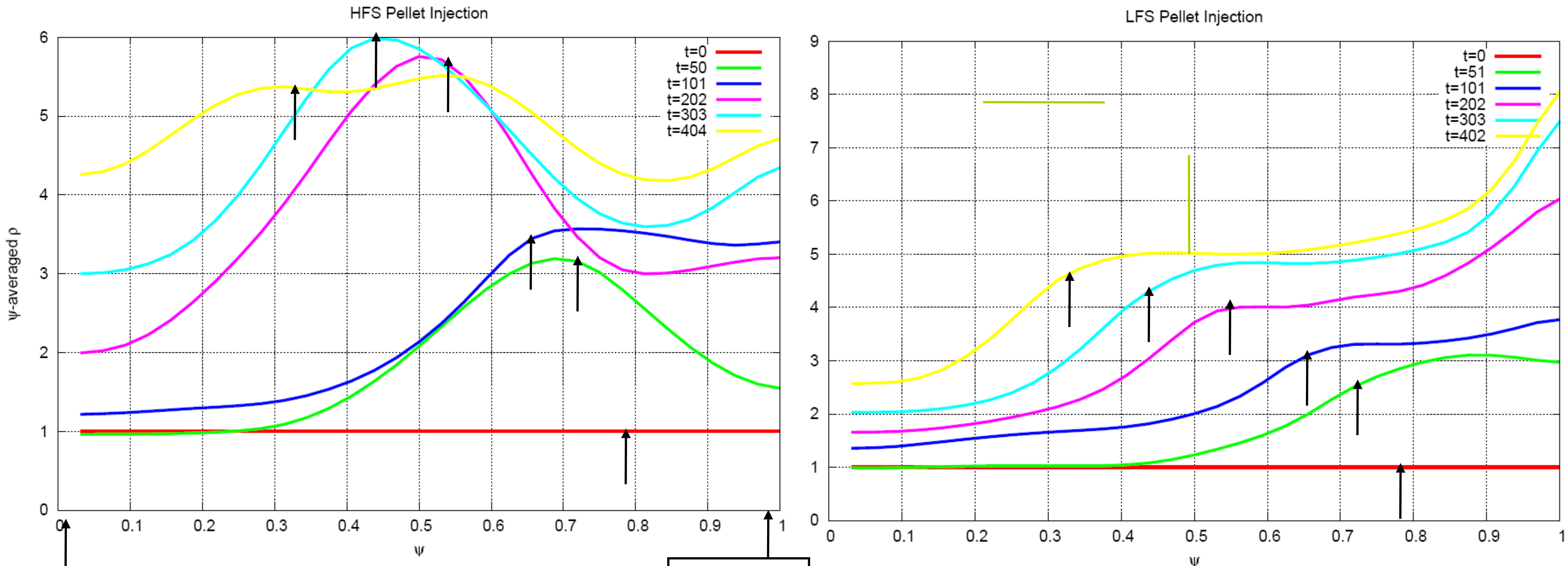
Pellet:

r_p=1mm,

v_p=1000m/s



HFS vs. LFS - Average Density Profiles



Core

Edge

HFS Pellet injection shows better core fueling than LFS

Arrows indicate average pellet location

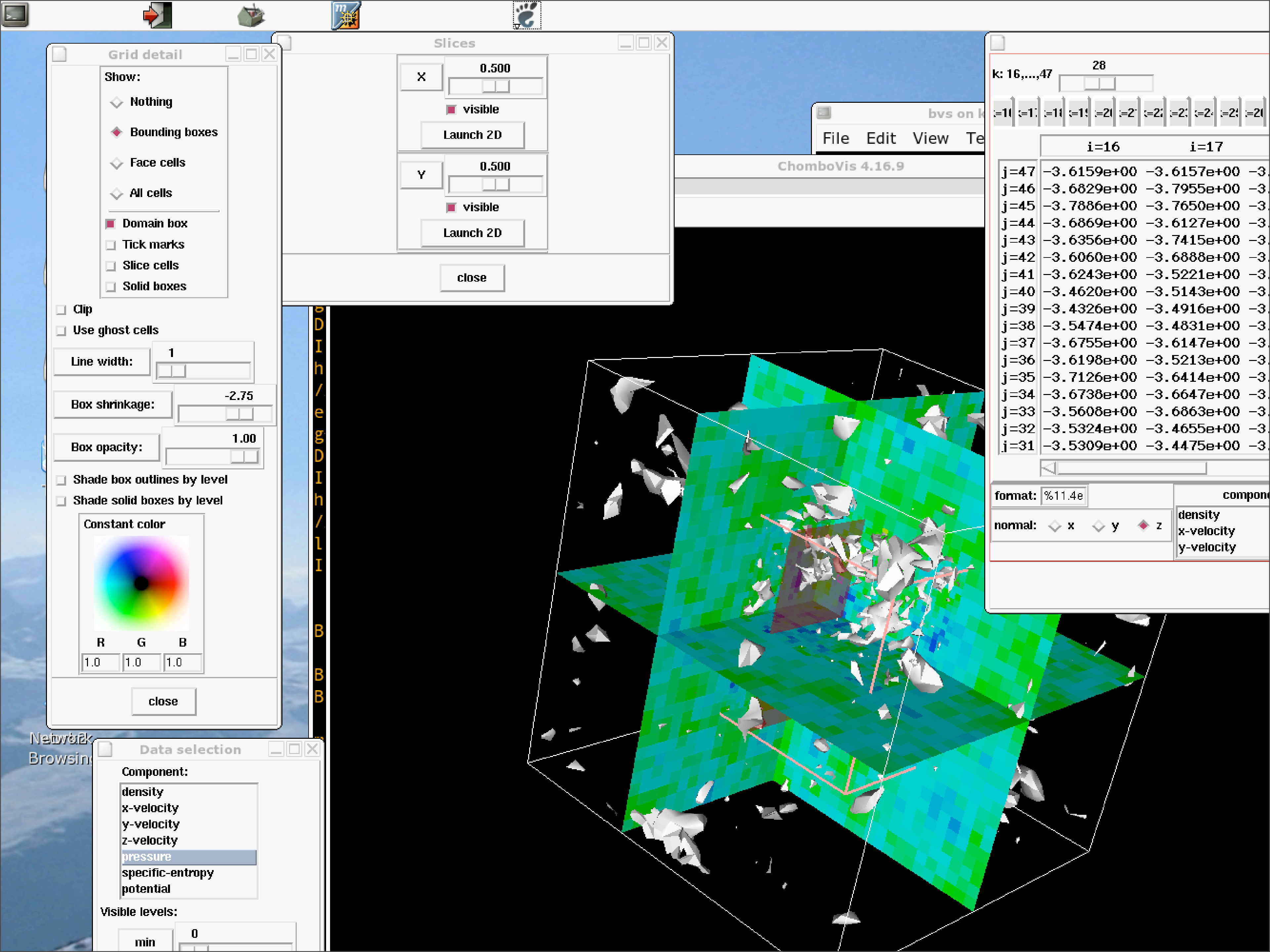
Visualization and Analysis

- Present/Past Visualization & Analysis
  - For Cartesian grids, ChomboVis was extensively used as the visualization tool
  - Derived quantities are regularly examined. Examples of derived quantities include:
    Current density J=curl(B), div(B) to examine divergence errors, J x B Lorentz force, magnetic pressure ($B^2/2$), kinetic energy
    - **Generally derived quantities are not easily computed with ChomboVis because of boundary conditions. The are computed at all plot times and written out.**
  - Several (x,y)-type time-history plots of quantities such as total energy (kinetic, thermal, magnetic), reconnection rates etc. These were done typically using post-processing of stored data
  - Slices (2D and 1D) along axes were plotted
  - For mapped grids, ChomboVis was used to examine the field variables in computational space. The field quantities were interpolated onto a single mesh and visualized using AVS

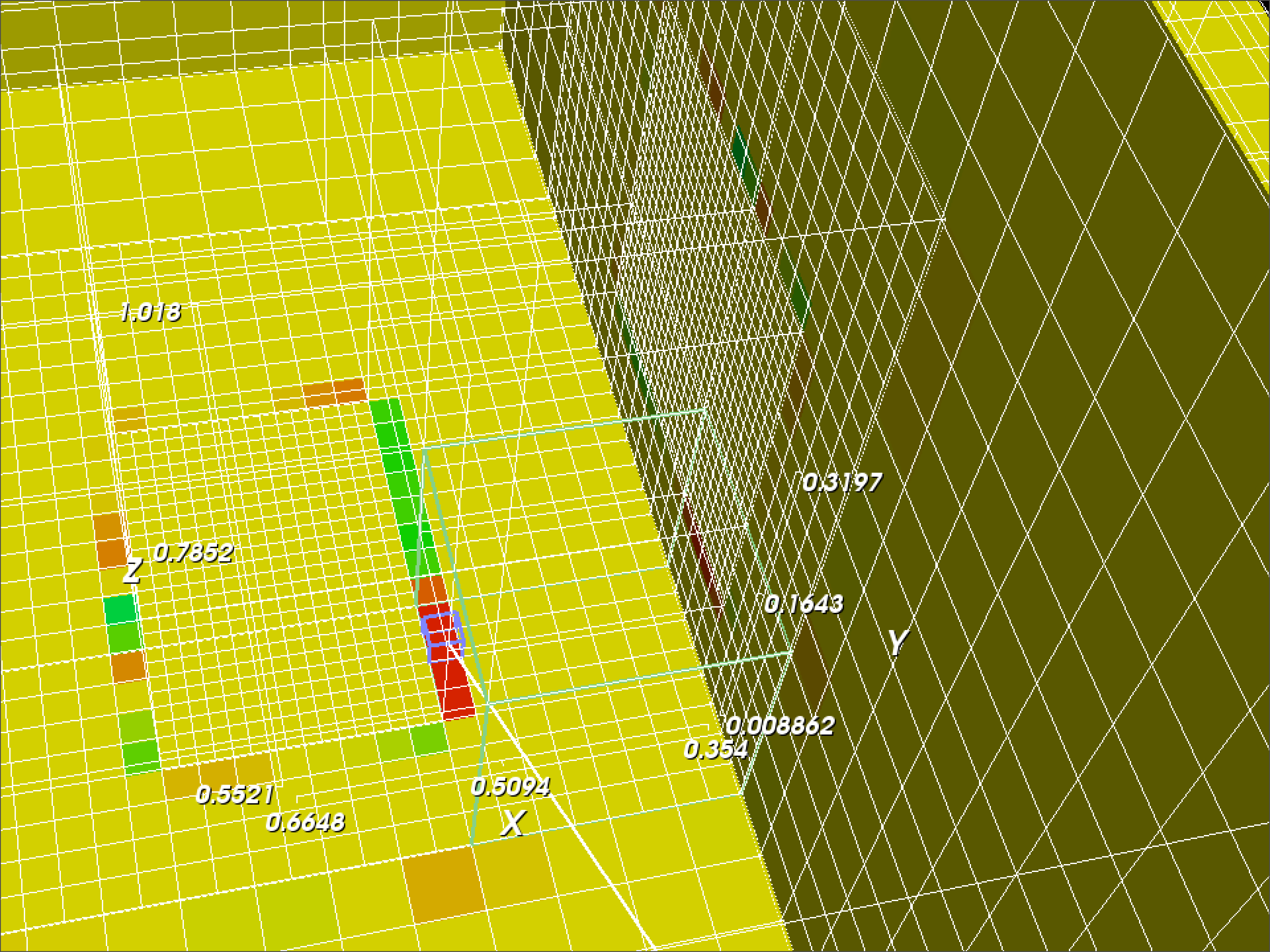
- **Future V & A needs for APDEC MHD**
  - Visualizations of field quantities on hierarchical mapped grids (isosurfaces, slices in 2D on arbitrary planes and 1D along arbitrary directions
  - Easy-to-use methods with ways to choose appropriate stencils and a subset of boundary conditions (e.g. zero gradient, reflecting etc.) to obtain a variety of derived quantities interactively (e.g. curl(B), curl(curl (B)), grad(div(V)) etc.)
  - Easy-to-use methods to examine vector field topology, and intuitive and easy interactive ways of picking seed points to generate and examine stream lines, path lines and streak lines, and get quantitative information at points along these streamlines such as local curvature
  - Volume rendering for hierarchical mapped and Cartesian grid data with

ChomboVis

- VTK + C++ classes and pipeline specifically modified to accommodate AMR data.
- Python interface and TKInter GUI
- Scripting
- Data analysis API
- Spreadsheet
- GDB invocation



Grid detail
Show:
Nothing
Bounding boxes
Face cells
All cells
Domain box
Tick marks
Slice cells
Solid boxes
Clip
Use ghost cells
Line width:
1
Box shrinkage:
-2.75
Box opacity:
1.00
Shade box outlines by level
Shade solid boxes by level
Constant color
R
G
B
1.0
1.0
1.0
close
Slices
X
0.500
visible
Launch 2D
Y
0.500
visible
Launch 2D
close
ChomboVis 4.16.9
File
Edit
View
k: 16,...,47
28
i=16
i=17
j=47 -3.6159e+00 -3.6157e+00
j=46 -3.6829e+00 -3.7955e+00
j=45 -3.7886e+00 -3.7650e+00
j=44 -3.6869e+00 -3.6127e+00
j=43 -3.6356e+00 -3.7415e+00
j=42 -3.6060e+00 -3.6888e+00
j=41 -3.6243e+00 -3.5221e+00
j=40 -3.4620e+00 -3.5143e+00
j=39 -3.4326e+00 -3.4916e+00
j=38 -3.5474e+00 -3.4831e+00
j=37 -3.6755e+00 -3.6147e+00
j=36 -3.6198e+00 -3.5213e+00
j=35 -3.7126e+00 -3.6414e+00
j=34 -3.6738e+00 -3.6647e+00
j=33 -3.5608e+00 -3.6863e+00
j=32 -3.5324e+00 -3.4655e+00
j=31 -3.5309e+00 -3.4475e+00
format: %11.4e
normal:
x
y
z
density
x-velocity
y-velocity
Data selection
Component:
density
x-velocity
y-velocity
z-velocity
pressure
specific-entropy
potential
Visible levels:
min
0


1.018
0.7852
Z
0.3197
0.1643
Y
0.008862
0.354
0.5094
0.5521
0.6648
X

ChomboVis Data API

class HDF5Dataset
HDF5Dataset(string infile_name);

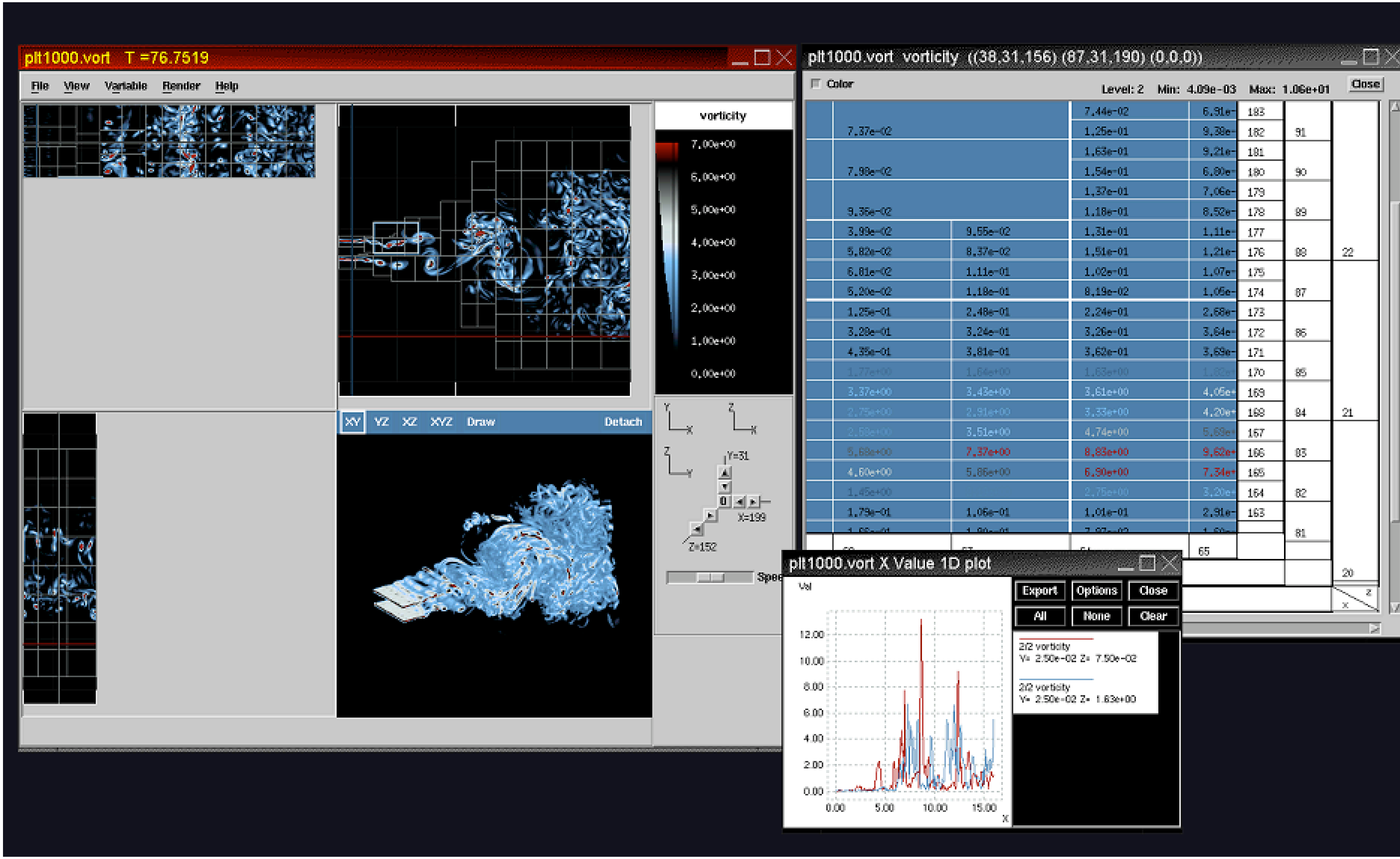
– HDF5Dataset(vector<vector<BoxLayoutData> >);
– vector getComponentNames() ;
– Box getProbDomain(int level) ;
– Realvect getDx(int level) ;
– Realvect getOrigin() ;
– Intvect getOutputGhost(int level) ;
– BoxLayout getBoxes(int level) ;
– **BoxLayoutData** const & getData(int level, string component);
- void defineNewComponent(string name, real (*function)(vector<real>), vector<string> arg_component_names);
– void defineNewComponent(string name, real (*function)(vector<real>), vector<int> arg_component_numbers);

Interpretive/Scripting ChomboVis API

```
>>> import chombovis

>>> hdf5_data = c.reader.getVTKData()

>>> def normFunc( x, y ): return pow( x*x + y*y, 0.5 )

>>> defineNewComponent( 'norm', normFunc, ('xnormal-0','ynormal-0'))

>>> print hdf5_data.getData( level=0, component='norm' ).getMax()
    0.0706233977

 >>> print c.reader.getDomainBounds()
{'z': (0.0, 0.03125), 'x': (0.0, 1.0), 'y': (0.0, 1.0)}
>>> import myutils

>>> myutils.diskAverageGUI( 'norm', a_level=0, a_center=(0.5,0.5) )

>>> hdf5_data.write( 'eb_with_norm.2d.hdf5' )
```

AMRVis

- Amrvis, AmrDerive, and AmrMovie
- 3D, 2D, 1D
- Slice along coordinate axes
- Spreadsheets
- Demand driven I/O
- Framework for “data browsing”
- Animations
- Distributed memory parallel
  - Interactive and batch modes.
  - Runs everywhere.



plt1000.vort T =76.7519
File
View
Variable
Render
Help
vorticity
7.00e+00
6.00e+00
5.00e+00
4.00e+00
3.00e+00
2.00e+00
1.00e+00
0.00e+00
XY
YZ
XZ
XYZ
Draw
Detach
Y=31
X=199
Z=152
plt1000.vort vorticity ((38,31,156) (87,31,190) (0,0,0))
Color
Level: 2
Min: 4.09e-03
Max: 1.06e+01
Close
plt1000.vort X Value 1D plot
Val
Export
Options
Close
All
None
Clear
2/2 vorticity
Y= 2.50e-02 Z= 7.50e-02
2/2 vorticity
Y= 2.50e-02 Z= 1.63e+00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
0.00
5.00
10.00
15.00
X

- **Limitations of Chombovis**
  - Currently unfunded
  - Serial
  - Only reads Chombo HDF5 files
  - Challenging to build
- **Limitations of AMRVis**
  - Currently unfunded
  - Limited 3D abilities (just moving slices)
  - Inscrutable internals
  - Only reads BoxLib file format
- **Limitations of Tecplot**
  - Poor scaling
  - Does not treat AMR data as first-class data structure

Requirements to date

- Visit reader for *Chombo HDF5* and *BoxLib* file formats
- .visitrc
  - Makes the user interface re-targetable to users with different existing tool familiarity.
  - Enable Migration
  - ChomboVis and AMRVis represent decades of use patterns for structured AMR computation.
  - Differences in how users wish the tool to start up
- Spreadsheets/DataBrowser
- Mapped AMR grids
- Embedded boundaries
  - Everything changes with EB.
  - EB + Combustion = **high impact**
- Some rational way of moving between workstation level and supercomputer level.
- Organized convergence studying. Currently we roll this by hand every time.

Requirements cont.

- Usable from the debugger
  - client-server ?
  - Create a vis tool analogue of "print" command for complex AMR data structures (more than just the field variables).
- Centerings:
  - Cell, Node, Face, Edge
- Scripting for time-dependent data
  - Currently ChomboVis exposes the python virtual machine to the user and has a Data API and Vis API
- Selecting IndexSpace regions
- User-supplied callback functions for data transformations.
  - Equations of state packages. Boundary conditions. Coarse-Fine issues
- Multi-block domains
- Control of Coarse-Fine handling.
- Volume Rendering
- AMR+FastBit+Visit+SRM+blah blah blah + kitchen sink

Requirements cont. …

- Performance …
  - Visualization tools will have to at least match existing tools to have a chance of meeting our future needs.
  - Need to exploit structured data in almost all operations.
    - How do we maximize performance in mapped and cut-cell data ?
  - Serial uniprocessor performance should not be neglected.
    - Code development vs. production runs